

REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-2, 4-6, 8-15, 17-20, and 22-25 pending in the application. The Examiner additionally stated that claims 1-2, 4-6, 8-15, 17-20, and 22-25 are rejected. By this communication, claim 2 is cancelled and claims 1, 17, and 22 are amended. Hence, claims 1, 4-6, 8-15, 17-20, and 22-25 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Specification

Applicant has amended the specification to secure a substantial correspondence between the claims amended herein and the remainder of the specification. No new matter is presented.

In the Claims

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claims 1-2, 4-6, 8-15, 17-20, and 22-25 under 35 U.S.C. 103(a) as being unpatentable over Kessler et al. (US 6,789,147), hereinafter, “Kessler,” in view of Christie et al. (US 7,165,135), hereinafter, “Christie.” Applicant respectfully traverses the Examiner’s rejections.

Claim 1 recites:

1. An apparatus for performing cryptographic operations, comprising:

an x86-compatible microprocessor, comprising:

an instruction register within a x86-compatible microprocessor having a single, atomic cryptographic instruction disposed therein, wherein said single, atomic cryptographic instruction is arranged according to the instruction format for execution on said x86-compatible microprocessor, and wherein said single, atomic cryptographic instruction is part of an application program, and wherein said x86-compatible microprocessor executes said application program, and wherein said single, atomic cryptographic instruction prescribes an encryption operation, and wherein said single, atomic cryptographic instruction prescribes that a user-generated key schedule be employed for execution of said encryption operation, and wherein said encryption operation that is prescribed by said single, atomic cryptographic instruction comprises encryption of a plurality of plaintext blocks to generate a corresponding plurality of ciphertext blocks;

a keygen unit, operatively coupled to said instruction register, configured to direct said x86-compatible microprocessor to load said user-generated key schedule; and

an execution unit, operatively coupled to said keygen unit, configured to employ said user-generated key schedule to execute said encryption operation, said execution unit comprising:

a cryptography unit, configured execute a plurality of cryptographic rounds on each of a plurality of input text blocks to generate a corresponding each of a plurality of output text blocks, wherein said plurality of cryptographic rounds are prescribed by a control word that is provided to said cryptography unit.

In the instant Office Action, the Examiner argues that Christie further discloses a single, atomic cryptographic instruction *that prescribes an encryption operation*, citing Christie col. 6, lines 35-53. This argument is not persuasive in that Christie discloses no such feature. For the Examiner's reference, the noted section of Christie is provided below.

35 this). These requirements may be met by using CPU micro-
code and hardware mechanisms that are unavailable to
normal software. A specific feature of the SEM-capable
processor, the Security Kernel Initialization (SKINIT)
40 instruction, is used to perform various security initialization
functions such as reinitializing the SEM-capable processor,
verifying that other processors in an MP system are stopped,
and communicating with the SSP using reserved system
addresses and unique bus transactions and jumping into the
SK code, for example. It is noted that in one embodiment,
45 the SKINIT instruction is a single instruction having multi-
ple microcode components that execute atomically. As
used herein, to execute atomically refers to executing to
completion in a specified order without interruption. It is
further noted that in one embodiment, the initialization
50 process includes two steps: the validation of a small Secure
Loader (SL) by the SSP, followed by the validation of the
much larger SK.

Applicant is receptive of the Examiner pointing out in the above section, or any other section of Christie, where a single, atomic cryptographic instruction *that prescribes an encryption operation*, is disclosed. However, based upon Applicant's thorough examination of the reference, it is clear that Christie's SKINIT instruction is used to perform various security *initialization* functions such as re-initializing the processor, verifying that other processors in the system are stopped, and communicating with the SSP using reserved system addresses and unique bus transactions. It is true that Christie's

SKINIT instruction executes atomically, but it is also true that his atomically executing instruction *does not prescribe an encryption operation*. And as one skilled in the art will concur, the complexity associated with a prescribed operation determines whether or not such an operation can be executed atomically. That Applicant has provided an apparatus and method in the instant application for prescribing an encryption operation via an x86 instruction, and furthermore providing for *atomic execution of the prescribed encryption operation on an x86-compatible microprocessor*, patentably distinguishes the present invention over any conceivable combination of the cited references.

The Examiner argues Applicant's assertion that the coprocessor of Kessler could be used to offload the x86-compatible microprocessor of Christie for the performance of security functions is not persuasive because cited portions of Kessler (col. 1, line 66 – col. 2, line 39) do not suggest that the purpose of Kessler's invention is to "offload tasks from the host processor, noting that Applicant merely pointed to the background of the invention where Kessler describes the use of coprocessors. In response, this argument is not persuasive because Kessler's invention itself is an interface for a security *coprocessor*, and thus his background discussion of coprocessors is entirely relevant to place his invention in context. Kessler's invention is clearly directed at offloading tasks from the host processor. Note in claim 1 Kessler recites retrieving . . . requests . . . from a host memory. In claim 6, Kessler limits this retrieving to being DMA reads. His clearest indication that a coprocessor performs the security operations is seen in claim 11, where a host processor stores a number of *requests* for security operations and a *coprocessor* completes the requests and writes the results to allocated data structures.

The Examiner further argues that Christie discloses a hash of Security Kernel to verify security service processor by citing col. 6, lines 35-53. This argument is not persuasive in because Christie discloses no such feature. Again, the cited section of Christie is provided below.

35 this). These requirements may be met by using CPU micro-
code and hardware mechanisms that are unavailable to
normal software. A specific feature of the SEM-capable
processor, the Security Kernel Initialization (SKINIT)
instruction, is used to perform various security initialization
40 functions such as reinitializing the SEM-capable processor,
verifying that other processors in an MP system are stopped,
and communicating with the SSP using reserved system
addresses and unique bus transactions and jumping into the
SK code, for example. It is noted that in one embodiment,
45 the SKINIT instruction is a single instruction having mul-
tiple microcode components that execute atomically. As
used herein, to execute atomically refers to executing to
completion in a specified order without interruption. It is
further noted that in one embodiment, the initialization
50 process includes two steps: the validation of a small Secure
Loader (SL) by the SSP, followed by the validation of the
much larger SK.

As can be determined from examination of the relevant passage, Christie does not even mention the word "hash." Moreover, Applicant submits that a hash operation is not an encryption operation. These two operations are not equivalent.

Nowhere does the cited art disclose **wherein said single, atomic cryptographic instruction prescribes an encryption operation**, as is recited in claim 1. Kessler certainly does not disclose any form of a single atomic cryptographic instruction, for Kessler fails to teach any form instructions for programming a microprocessor. As has been previously submitted, Kessler teaches a security co-processor interface.

Christie indeed discloses an instruction that is atomic, and that is executed on an x86-compatible microprocessor, and wherein said x86-compatible microprocessor executes an application program. However, Christie entirely fails to teach a single, atomic cryptographic instruction that prescribes an encryption operation. Christie's SKINIT instruction is admittedly atomic, however it does not prescribe an encryption operation. Rather, Christie's instruction is used to perform various initialization functions in a secure execution mode capable microprocessor, and Christie's microprocessor is entirely incapable of performing an encryption operation.

Applicant fails to appreciate how Kessler and Christie can be combined in any practical manner to yield the limitation cite above. The references in combination fail to suggest, or allude to a single, atomic cryptographic instruction that prescribes an encryption operation.

By combining the references, one skilled in the art would be led to conclude that the coprocessor of Kessler may be useful in an x86 environment because it could offload cryptographic functions which would otherwise have to be performed via operating system intensive subroutine calls. That is, one skilled would appreciate that the coprocessor of Kessler could be used to offload the x86-compatible microprocessor of Christie for the performance of security functions, for Christie's processor is entirely void of any encryption capability. Christie's microprocessor is employed to provide a secure execution environment for trusted application programs, not to perform cryptographic operations such as encryption or decryption of message blocks.

As is pointed out in the instant specification, message encryption and decryption are very commonly employed operations, and there has been a noted desire in the community to accelerate these operations because they have heretofore been performed via either dedicated calls to software subroutines or by co-processors such as that taught by Kessler. And there are numerous manufacturers of x86-compatible microprocessors to include Intel Corporation and Advance Micro Devices. Yet, *none* of these manufacturers have been able—to date—to develop an x86-compatible microprocessor that meets the elements and limitations recited in claims 1, 17, and 22. Accordingly, it is respectfully asserted that one of ordinary skill in the art at the time of the invention would have not been informed by the combination of Kessler and Christie to yield the elements and limitations of the independent claims.

Nowhere does the cited art disclose **wherein said single, atomic cryptographic instruction prescribes that a user-generated key schedule be employed for execution of said encryption operation**, as is recited in claim 1. Both references are *utterly silent* with regard to use of a user-generated key schedule for employment in an encryption operation. Consequently, they do not teach, suggest, or event hint that a single, atomic

cryptographic instruction prescribes that a user-generated key schedule be employed for execution of said encryption operation.

Again, regarding the combined teaching of the cited references, Applicant fails to understand how they could be combined in any practical manner to yield the elements and limitations recited in claim 1. That is, Kessler explicitly teaches that a security operation *cannot* be performed by a general purpose microprocessor (i.e., host processor) and must be offloaded to a coprocessor—a *technique which is highlighted in the background of the instant specification as being limiting and disadvantageous*. Christie teaches a specific atomic instruction that is employed to initialize a secure mode in an x86 microprocessor.

Again, none of the cited references teach a cryptography unit disposed in an x86-compatible microprocessor.

The references in combination do not teach an x86-compatible microprocessor, nor does the combination teach a single, atomic cryptographic instruction (i.e., a macro instruction) within an application program that directs an x86-compatible microprocessor to perform an encryption operation. This is because, prior to the advent of the present invention, x86-compatible microprocessors could not be programmed via a single instruction to execute a cryptographic operation. There was no instruction, nor was there a cryptographic unit therein capable of performing the operation.

It is thus respectfully submitted that the combination of Kessler and Christie does not contemplate or suggest an x86-compatible microprocessor that includes a cryptographic unit within its execution stage. Hence, it further does not follow that the references, in combination would suggest a single, atomic cryptographic instruction which is part of an application program being executed by the x86-compatible microprocessor, and which directs that the microprocessor perform an encryption operation. It is respectfully submitted that any microprocessor that lacks a cryptographic unit therein would be incapable of performing the cryptographic function directed by the single, atomic cryptographic instruction. A skilled artisan would be forced to employ the technique taught by Kessler (i.e., offload the hash to a coprocessor) or the software technique

taught by other references (i.e., execute hundreds of macroinstructions (“software modules”) to perform the operation on a general purpose microprocessor—both techniques of which are shown to be inferior to the present invention.

Thus, for at least these reasons, it is respectfully submitted that the invention of claim 1 is patentably distinct and non-obvious in view of the cited art.

Claim 17 recites limitations similar to claim 1 with the exception that the single, atomic cryptographic instruction prescribes a decryption operation instead of an encryption operation.

Claim 22 recites substantially similar limitation as are recited in claim 1.

Accordingly, it is requested that the rejections of claim 1, 17, and 22 be withdrawn.

By this communication, claim 2 is cancelled, thereby rendering the rejection moot.

With respect to claims 4-6 and 8-15, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by the combination of Kessler and Christie. Accordingly, Applicant respectfully requests that the Examiner withdraw the rejections of claims 4-6 and 8-15.

With respect to claims 18-20, these claims depend from claim 17 and add further limitations that are neither anticipated nor made obvious by the combination of Kessler and Juffa. Accordingly, Applicant respectfully requests that the Examiner withdraw the rejections of claims 18-20.

Claims 23-25 depend from claim 22 and add further limitations that are neither anticipated nor made obvious by the combination of Kessler and Juffa. Accordingly, Applicant respectfully submits that claims 23-25 are allowable as well.

CONCLUSIONS

Applicant believes this to be a complete response to all of the issues raised in the instant office action and further submits, in view of the amendments and arguments advanced above, that claims 1, 4-6, 8-15, 17-20, and 22-25 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant also notes that any amendments made by way of this response, and the observations contained herein, are made solely for the purpose of expediting the patent application process in a manner consistent with the PTO's Patent business Goals (PBG), 65 Fed. Reg. 54603 (September 8, 2000), and are furthermore made without prejudice to Applicant under this or any other jurisdictions. It is moreover asserted that insofar as any subject matter might otherwise be regarded as having been abandoned or effectively disclaimed by virtue of amendments made herein and/or incorporated in attachments submitted with this response, Applicants wishes to reserve the right and hereby provides notice of intent to restore such subject matter and/or file a continuation application in respect thereof.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

Respectfully submitted,
HUFFMAN PATENT GROUP, LLC

/ Richard K. Huffman /

By: _____

RICHARD K. HUFFMAN, P.E.
Registration No. 41,082
Tel: (719) 575-9998

04 / 22 / 2010

Date: _____